Generating adaptive security policies and automated configuration scenarios in intrusion management systems

Dimitrios Patsos, Sarandis Mitropoulos, Christos Douligeris

Department of Informatics, University of Piraeus,
80, Karaoli and Dimitriou Street, 185 40 Piraeus, Greece.
{dpat, sarandis, cdoulig}@ unipi.gr

Abstract

Modern security technologies provide a large set of tools for automated vulnerability assessment (VA) and Intrusion Detection and Prevention (IDP). This paper after a brief exploration of the benefits and limitations of these technologies, introduces the concept of Intrusion Management Systems (IMS) that exchange, correlate and validate valuable security information. IMS combine, complement and leverage the effectiveness of the aforementioned techniques. We propose the use of IMS for the automated generation of adaptive security policies as well as the enforcement of these policies to IDP and VA technologies, via well-defined configuration scenarios. Finally, we highlight implementation issues of the IMS, discuss the benefits of our approach to post-incident procedures, like Incident Response and Digital Forensics, and address open issues and limitations of the current proposals.

Keywords: Intrusion Detection, Vulnerability Assessment, Intrusion Management Systems, Incident Response
Designated track: Research

1. Introduction

By nature, one or more potential hazards exist in nearly every piece of software. Any such potential hazard is known as vulnerability. A number of research efforts have tried to classify and model vulnerabilities, like [Rodgers et. Al. (2001)] and [Hansman (2003)], but –up to now and to the best of our knowledge- no formal or standard method exists to achieve this goal. In addition to this, the research community has not yet defined a standardized language to describe vulnerability semantics [Kumar et. al. (2005)]. As an example, a number of celebrated vulnerability reporting lists include Carnegie Mellon's CERT/CC (www.cert.gr), Mitre Corporation's CVE (cve.mitre.org), Symantec's Security Focus (www.symantecfocus.com) and Bugtraq which can be found on the Web. Moreover, vendors like Microsoft, Cisco, and Oracle maintain vulnerability lists for their products. Moreover, a large number of

Computer Response Teams and Analysis Centres dispersed all over the world maintain and publish to a large extent the same vulnerability information [Schultz (2004)]. Finally, both the open-source community and the security vendors develop vulnerability assessment tools (e.g. Nessus, ISS Internet Scanner, etc.) than contain vulnerability information and point to some of the aforementioned lists.

Tightly linked with vulnerabilities are the exploits. The exploits comprise the pragmatic information (e.g. piece of code) that utilizes one or more vulnerabilities to realize an actual attack. An attack path is defined as the series of consecutive vulnerability exploits that result in the realization of an attack. An attack path a_i can be formally defined as a unique sequence of successful vulnerability exploits[1], that is $a_i = \{(e_{j1}|v_{k1}), (e_{j2}|v_{k2}) ,\ldots, (e_{jp}|v_{kq})\}$, where $a_i \in A, \forall\, i \in N$, $e_{jp} \in E$, $\forall\, j,p \in N$, and $v_{kq} \in V, \forall\, k,q \in N$, A is the set of attack paths a_i, E is the set of exploits e_{jp} and V is the set of vulnerabilities v_{kq}. A formal presentation of attack paths can be found in [Krasser et. al. (2005)].

It is obvious that, there is a many to many relationship between vulnerabilities and exploits, that is one exploit can utilize one or more vulnerabilities and vice versa. However, what is of importance is that the actual significance of a vulnerability can be estimated only when the environment a particular vulnerability resides within is given. Information regarding exploits or the actual exploits are more difficult to be found, but there is a large number of websites known to host such information. These sites are usually maintained by security researchers or security companies (e.g. eEye, www.eEye.com) or can be maintained by hacking communities. Most of the times, the information related to exploits is also linked with specific vulnerabilities. Last but not least, there are also some websites of the so-called "black-hat" community that maintain complete attack path information, i.e. the actual series of vulnerability/exploit combinations that can help an attacker reach his objectives.

To defend against an attack, an attack is usually addressed by one (or more) corresponding signatures[2], which are usually found in antivirus programs or IDP systems. A signature contains the exploit code itself or, more frequently, a

[1] We assume, without real loss of generality, that an attack path is formed only when vulnerabilities are successfully exploited in a predefined order and that a different order does not result to the same attack path, if it even results to any. The proof of this concept is beyond the scope of this paper.

[2] The term "signature" can be used to describe either antivirus definition files or IDP patterns and variances of known attacks.

synopsis of the exploit code. In general, most IDP systems are loaded with a large set of such signatures and compare every packet intercepted against every one of these signatures (or according to what the security policy indicates). The obvious drawback one may see is that, without properly defined policies, the effectiveness of IDP systems is rather low, especially in networks with heavy traffic, while –as new signatures are loaded- the IDP resources are being exhausted.

A major issue in this context is the mapping of vulnerabilities to exploits on one hand, and exploits to signatures on the other hand. This will result in the construction of attack paths and counter-attack paths respectively. The attack path, as mentioned before, will contain the linkage between vulnerabilities and exploits, while the counter-attack path will contain the necessary signatures to address these exploits. This approach can provide interesting configuration scenarios for IDP systems. One such scenario can be based on security policies that adapt according to every specific attack path. This idea is the main research driver for the introduction of the Intrusion Management Systems (IMS) that are explained in the following sections. The IMS main objectives are to reduce IDP mechanisms false positives, eliminate VA false negatives, as well as considerably increase the policy effectiveness of IDP systems, by applying adaptive security policies in every counter attack path constructed. IMS exchange, correlate and validate security information, as well as combine, complement, and leverage the effectiveness of a number of well known techniques.

2. Related Work

Templeton and Levitt have provided a flexible model for computer attacks along with several proposed applications in VA and IDP [Templeton et. al (2000)]. Swiler et.al have developed an automated tool capable of generating and analyzing attack path information [Swiler et. al. (2001)]. Sheyner et. al. have also provided a tool that correlates attack graphs with the most exploitable components of the system configuration [Sheyner et. al (2002)]. Ammann, et. al. provide a scalable representation of attack graphs, focusing on revealing end-to-end attack scenarios [Ammann et. al (2002)]. None of these works, however, implements matching scenarios with IDP policies, since the primary focus is the modelling of network and computer-based attacks, as well as the production of attack paths and/or graphs.

Gula has highlighted various configuration scenarios where vulnerability information could be correlated with IDP audit log information, as an effort to reduce false positive information provided by IDP systems [Gula (2002)]. Nevertheless, this work does not include "filtering" mechanisms on the information used as input both in vulnerability assessment tools, as well as in Intrusion Detection Systems. Ning and Xu have developed a number of techniques for the automatic learning of attack strategies from intrusion alerts [Ning et. al. (2003)]. This work provides an effort for the effective correlation of IDP information with static VA information used as input, but it does not model network security conditions and/or analyze the attack paths. This is described by Jajodia et al in [Kumar et. al. (2005)] in terms of the Topological Vulnerability Analysis (TVA), a technique heavily used in the framework of IMS.

3. Intrusion Management Systems

The modern automated VA tools cannot identify the security policies that are enforced by the security mechanisms that in place at the segment assessed [Templeton et. al. (2000)]. Due to this, the output of a VA can differ, based upon where –in the network topology- the scan is performed. For example, if no security mechanism is placed between an assessment tool and an un-patched Web Server, the results will be entirely different from those of a scan where a properly configured firewall is placed between the assessment tool and the Web Server. The reason is that the security policy enforced by the firewall limits the server's responses to the potential connections requested by the scanner. The oxymoron is that, in both cases, exactly the same vulnerabilities exist on this Web Server. Due to this limitation, the assessment tools cannot construct attack paths. Besides, the identification of a large a number of vulnerabilities in a network segment or a system does not indicate a high security exposure of this segment or system, since it is possible that the vulnerabilities unearthed cannot be exploited with a predefined order or a combination that leads to an actual attack. Therefore, they do not correspond to exploitable attack paths. The latter is mainly performed manually (or semi-automatically in the best case) by highly-skilled security analysts who arre able to understand the security context of a discovered vulnerability.

On the other hand, IDP systems have significantly advanced in last years so that they are able to be deployed in a variety of topological elements or network devices. For example, apart from dedicated devices, they can be

integrated in edge routers, in application firewalls, and in endpoint security solutions. IDP systems are based upon signatures that contain the actual exploit/attack code or, more frequently, a synopsis of the exploit/attack code. It must be also noted that IDP systems aim at the prevention or detection if attacks (not only vulnerabilities).

The gap between the security information provided by VA tools and the information need to enforce effective and efficient IDP policies can be greatly reduced by the use of Intrusion Management Systems (IMS). An IMS is a security management system, which has the following capabilities:

- Construction of attack paths, using VA information as input (by producing a real-time vulnerability-exploit mapping).
- Construction of counter-attack paths, using IDP information as input (by a real-time exploit-signature mapping).
- Attack path and counter attack path correlation, and aggregation as well as construction of a real-time vulnerability, exploit and signature mapping.
- Generation of per attack path policies.
- Enforcement of per attack path policies to IDP by issuing appropriate policy commands.
- Enhancements to Incident Response and Digital Forensics.

An IMS comprises of several modules as depicted in Fig.1. A brief description of every module follows.

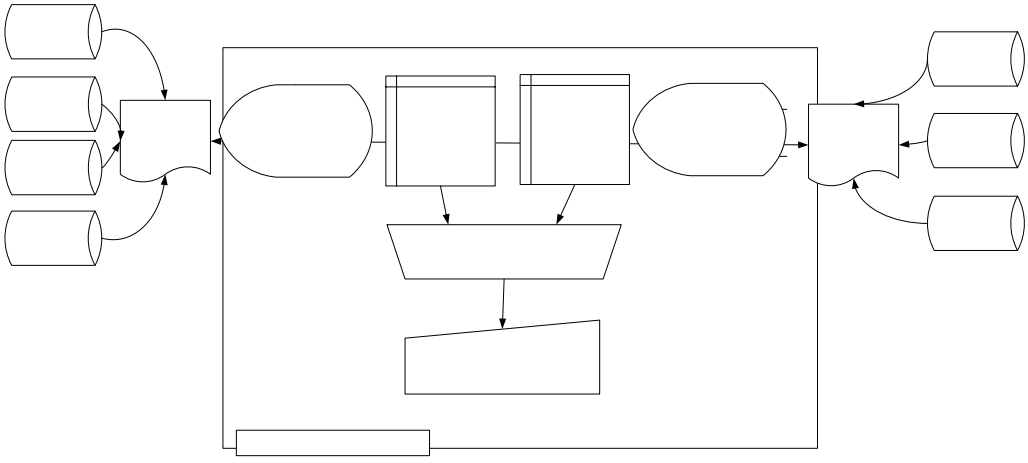

Figure 1. *A high level overview of an Intrusion Management System (IMS)*

3.1 Construction of attack paths

In order to construct an attack path, the IMS modules are performing the following tasks:

- The Vulnerability Gathering Module (VGM) uses a process to gather vulnerability information from various sources (e.g. the Web, mailing lists, VA tools, etc.).
- The Vulnerability Exploit Extraction Module (VEEM) uses a process to gather exploit information from various sources (e.g. security research sites, hacking sites, etc.).
- An XML program links vulnerabilities with exploits, producing a real-time vulnerability/exploit mapping.
- The Vulnerability Storage Module (VSM) uses a process to store all the above information in the Vulnerability Information Base (VIB).

3.2 Construction of counter-attack paths

A similar process that deals with exploits and IDP signatures is performed in parallel. As mentioned earlier, a good source of such information is the IDP signature database that contains the exploit code. In order to construct a counter attack path, the IMS modules perform the following tasks:

- The Exploit Gathering Module (EGM) uses a process to gather attack information from various sources (e.g. the IDP database).
- The Signature Exploit Extraction Module (SEEM) uses a process to extract the exploit information found in these signatures.
- An XML program links signatures with exploits, producing a real-time signature/exploit mapping.
- The Vulnerability Storage Module (VSM) uses a process to store all the above information in the Exploit Information Base (EIB).

3.3 Attack and counter attack path correlation and aggregation and construction of a real-time vulnerability, exploit and signature mapping

When these two mentioned before processes are completed, several vulnerability/exploit and exploit/signature mappings are produced. These mappings have to be filtered and aggregated for redundant or not relative entries. What is important for an IMS is to enforce a security policy p_i in the IDP systems that will use certain signatures (e.g. $s_1,\ldots,s_n$) to address the vulnerabilities and exploits of the attack path a_i. A simplified expected output

is depicted in Fig. 2, where signatures $s_{1,}$ s_n belong in the intrusion detection policy domain p_1, which ideally counters the attack path $a_{1,.}$

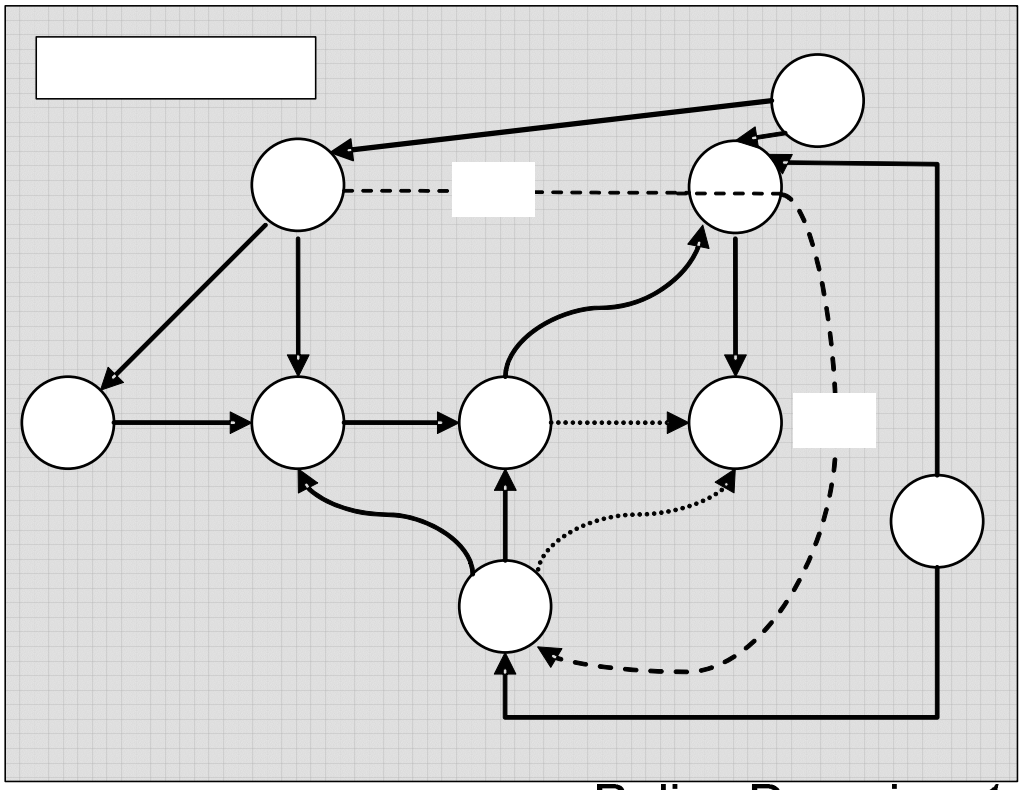

Figure 2. *Vulnerability, Exploit mapping within a policy domain*

It is expected that no 1-1 relationship should exist between attack paths and counter attack paths, but an entire attack path should correspond to a number of different IDP signatures that address this path (which correspond to a *per attack path policy*). Additionally, minimum false positives should normally exist in this phase, for attack paths have been discovered and only the corresponding IDP signatures have been activated.

3.4 Generation of per attack path policies and enforcement of per attack path policies in IDP

One of the most critical modules of an IMS, is the Policy Enforcement Module (PEM). PEM which is responsible for analyzing an existing IDP policy, as well as for adapting these policies to the needs of the attack and counter attack paths identified. In other words, the PEM identifies and selects only the

minimum set of signatures needed to counter a specific attack path. Moreover, it has to activate these signatures and reconfigure the IDP system. The latter can be done by using the SISL language (as defined in [Feiertag, 1999]), or another standardized language capable of issuing appropriate policy commands to the IDP. At this stage, it is expected that only one policy p_i corresponds to the attack path a_i, that is $p_i = (a_i \mid s_i)$, $\forall\, p_i \in P$, $\forall\, a_i \in A$, $\forall\, s_i \in S$, and $\forall\, i \in N$, where P is the set of *per attack path policies*.

The adaptive security policies can facilitate very flexible configuration scenarios in IDP systems, since the policies can be changed according to what the IMS indicates for a specific attack path. This feature is extremely useful when an attack –not addressed by any policy- is in progress, since the IMS can provide "self-resisting" attributes to the IDP by continuously modifying a generic baseline policy to counter the attack in progress.

3.5 Enhancements to Incident Response and Digital Forensics

Incident Response is the process of efficiently handling and responding to a security incident. As a corporate process, it was not -until recently- included or defined in hardly any formal information security standard [ISO (2005)]. Various formal methodologies on Incident Response propose manual or semi-automated procedures on identifying the incident's source, magnitude and severity so that decisions be taken. These decisions affect the members of nearly the entire scope of an organization, since a large set of company members have to take specific actions [Mitropoulos et. al. (2006)]. Thus, a critical part of the Incident Response process is the proper and timely identification of a security incident. Today, a large part of this procedure is carried out by high-end management systems (Security Information Management Systems (SIMs) that produce results based on correlating security information found in system, network, and application logs.

Intrusion Management Systems aim of eliminating false positive information provided by nearly all modern IDP technologies [Aberdeen (2003)], therefore providing more accurate information on an incident's occurrence. Moreover, their ability to produce adaptive security policies and issue the corresponding configuration commands to the IDP systems advise the Incident Response parties to adjust the policy according to the incident's characteristics. In other words, the information provided by SIMs can be used from the IMS as well, in terms of policy adjustment.

Furthermore, there are many cases where an organization decides to pursue a Digital Forensics analysis, so that the responsible party is held accountable. In this case, the IMS can provide the experts with the definition of the overall attack context (since it constructs the attack path), an issue of major importance when a forensics analysis is performed in "live" systems [Adelstein (2006)].

3.6 Implementation Details and Workflow of Operation

An Intrusion Management System is divided into 5 main layers and several integral modules, as depicted in Fig. 3. The operations of these layers and modules are described in this section.

The ***IMS Application Layer*** provides the system's interface to the end users and/or developers, by hosting the IMS APIs. Using these APIs the vulnerability, exploit and IDP signature sources can be defined. A number of various other system settings (e.g. system configuration, system update, etc) can be also configured.

Below the IMS Application Layer is the ***IMS Feature Gathering and Extraction Layer*** which facilitates information gathering and extraction. It contains the Vulnerability Gathering Module (VGM), the Vulnerability Exploit Extraction Module (VEEM), the Signature Exploit Gathering Module (EGM) and the Signature Exploit Extraction Module (SEEM).

The Vulnerability Gathering Module (VGM) gathers vulnerability information from sources that have been previously defined in an IMS API. This information can include Web Sites, mailing lists or input from VA tools. This information is then passed to the Vulnerability Exploit Extraction Module (VEEM) where all the information linked with vulnerabilities and exploits is extracted. Parallel to this linking, the Signature Exploit Gathering Module (EGM) is also gathering exploit information from other sources that have been previously defined in another IMS APIs (such as Web Sites, mailing lists or input from IDP). The Signature Exploit Extraction Module (SEEM) is responsible to extract all the information regarding the relevancy between exploits and signatures.

The ***Real-Time Mapping and Constructing Layer*** provides the real-time mapping between vulnerabilities, exploits and signatures, by constructing the appropriate attack and counter attack paths. This layer contains the Vulnerability/Exploit Mapping Sub-module that uses the VEEM input to

construct the real-time mapping between vulnerabilities and exploits. This sub-module provides a Topological Vulnerability Analysis graph that contains the attack paths derived from the existing vulnerabilities. This layer also contains the Exploit/Signature Mapping Sub-module that uses the SEEM output in order to construct a real-time mapping between exploits and signatures, providing a topological analysis of the infrastructure's current defences. The Attack path constructing sub-module is also part of this layer constructing the corresponding attack paths.

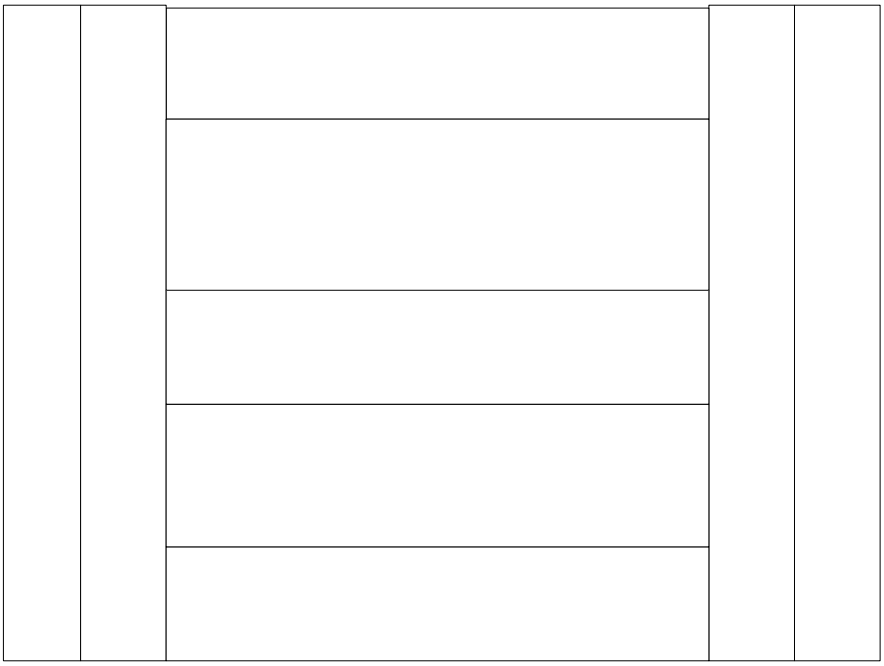

Figure 3. *The IMS Architectural layers*

The **Policy Construction and Enforcement Layer** is responsible for the generation of adaptive security policies as well as for enforcing these policies by the issuance of the appropriate commands. It contains the secure storage sub-module, where the outputs of the Vulnerability/Exploit Mapping Sub-module and the Exploit/Signature Mapping Sub-module are stored in the Vulnerability Information Base (VIB) and the Exploit Information Base (EIB) respectively, the Correlation engine that cooperates with a set of Policy construction tools that contain correlation rules to normalize and, in turn,

correlate the vulnerability/exploit and exploit/signature mappings, a set of Conflict resolution rules and tools, the Consistency analysis module and the Policy Enforcement Module (PEM) that define error-free and appropriate per attack path policies to the IDP systems managed by the IMS.

The bottom layer of the IMS tiers is the **Infrastructure Layer** that supports the necessary operating system, network and RDBMS needs of the IMS. Moreover, a large set of system health monitoring tools guarantee the smooth operation of all IMS components. In this layer, various auditing and accounting tools are responsible for producing reports. Finally, the IMS is supported by appropriate management tools for the necessary security updates of the IMS components.

3.7 Limitations

An IMS is a management system, so their results are still based on the capabilities of VA tools and IDP systems. In other words, an IMS cannot assist in cases when the VA tool misses the detection of a vulnerability or an IDP system identifies normal traffic as an attack. If something like this happens, it is quite likely that the IMS results will not be accurate.

Furthermore, the fact the security research community is currently missing a standardized predefined format for both vulnerability and attack description [Gordon (2003)], a notable obstacle exists for the IMS capability of understanding vulnerability information found in proprietary or commercial tools. In the world of IDP mechanisms the same obstacle exists: up to now there is no a predefined standard for IDP signatures. These limitations are some of the current main development barriers, since the development of IMS can be only based upon *reference* systems like the open-source Snort Intrusion Detection System (www.snort.org) and the Nessus vulnerability scanner (www.nessus.org).

Finally, the issue of exchanging information between VA tools and IDP systems in a way that one system provides feedback to the other is not yet effectively addressed. Up to now, only commercial products of the same vendor can provide this functionality in a rather limited way.

4. Conclusions and Future Work

This paper briefly explored the limitations of automated Vulnerability Assessment (VA) tools and Intrusion Detection and Prevention (IDP) systems and highlighted the fact that these technologies cannot operate in isolation. We subsequently introduced the concept of Intrusion Management Systems (IMS) that exchange, correlate and validate valuable security information and which, in turn, combine, complement and leverage the effectiveness of the aforementioned techniques. Furthermore, we proposed the use of IMS for the automatic generation of adaptive security policies and the enforcement of these policies to IDP systems and VA tools, via well-defined configuration scenarios. Finally, we proposed an implementation approach for IMS, discussed the benefits of our approach to post-incident procedures, like Incident Response and Digital Forensics, and highlighted open issues and current IMS development limitations.

Our next immediate research steps are to finalize the development of an entire IMS, based upon reference legacy systems (VA and IDP). Moreover, a proposed schema for the vulnerability and intrusion information standardization is also in progress to assist in bypassing this major obstacle and facilitate future growth in IMS development.

Acknowledgements

This paper has been partially supported by GSRT under a PENED grant and by the IST FET Coordination Action ACCA 6475).

5. References

Aberdeen Group (2003), Turning IT Security into Effective Business Risk Management, An Executive White Paper. Available at: http://www.ca.com/

Adelstein, F. (2006), Live Forensics: Diagnosing your system without killing it first, *Communications of the ACM, Vol. 49, No. 2*

Ammann, P., Wijesekera, D., Kaushik S., (2002), "Scalable, Graph-Based Net-work Vulnerability Analysis," *in Proceedings of CCS 2002: 9th ACM Conference on Computer and Communications Security, Washington, DC, p.p. 217 - 224.*

BSI (1999), Information Security Management, BS7799, Part 1: Code of Practice for Information Security Management

Feiertag R., Kahn C., Porras P., Schnackenberg D., Staniford-Chen S. and Tung B. (1999), "A Common Intrusion Specification Language"

Gordon, S. (2003), Virus and Vulnerability Classification Schemes: Standards and Integration, Symantec Security Response (White Paper), Symantec Corporation, Available: www.symantec.com

Gula, R. (2002), Correlating IDS Alerts with Vulnerability Information, Tenable Network Security, Available: www.tenablesecurity.com

Hansman, S. (2003), A Taxonomy of Network and Computer Attack Methodologies, Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand

IEEE (2003), IEEE Std. 802.1Q-2003, Virtual Bridged Local Area Networks; ISBN 0-7381-3662-X

ISO/IEC FDIS 27001 (2005), International Standard, Information Technology – Security Techniques – Information Security management Systems – Requirements, Geneva

Krasser, S., Conti, G., Grizzard, J., Gribschaw, J., Owen, H. (2005), Real-Time and Forensic Network Data Analysis Using Animated and Coordinated Visualization, *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, p.p. 42-49.*

Kumar, V., Srivastava, J., Lazarevic, A., (eds.) (2005), Managing Cyber Threats: Issues, Approaches and Challenges, *Massive Computing Series, Vol. 5, p.p.247-267, Springer Verlang*

Mitropoulos, S., Patsos, D., Douligeris, C. (2006), On Incident Response Categorization: A State of The Art Approach, *Computers and Security, Vol. 25, Issue 5, pp. 351-37, Elsevier*

Ning, P. and Xu, D. (2003), Learning attack strategies from intrusion alerts, *In Proceedings of the 10th ACM conference on Computer and communications security, Washington D.C., USA, pp.200-209*

Rodgers, C., Hunt, R., Harris. B, (2001), Networking Systems - Design, Analysis and Applications: Threats to TCP/IP Network Security, COSC407 Research Project

Schultz, E. (2004), Incident Response Teams Need to Change, *Computers and Security, vol. 23, pp. 87-88.*

Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing J., (2002), "Automated Generation and Analysis of Attack Graphs," *Proceedings of IEEE Symposium on Security and Privacy, Oakland, California, p.p. 273- 284*

Swiler, L. P., Phillips, C., Ellis, D., and Chakerian, S., Computer-Attack Graph Generation Tool, *DISCEXII Proceedings, DARPA's Information Survivability Conference and Exposition, IEEE Computer Society Press, Vol. 2, p.p. 307-321*

Templeton, S., Levitt, K. (2000), "A Requires/Provides Model for Computer Attacks", Proceedings *of the New Security Paradigms Workshop, Ballycotton, County Cork, Ireland, p.p. 31-38.*